

SKRIPSI

**PERANCANGAN PENGATURAN KECEPATAN
MOTOR INDUKSI SATU FASA DENGAN PWM
MENGUNAKAN PENGENDALI PID BERBASIS
ARDUINO**

**Diajukan untuk memenuhi persyaratan dalam menyelesaikan
Pendidikan Tingkat Sarjana (S1)**



Oleh :

TIRTA WISESA

G1D009060

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2014**

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

PERANCANGAN PENGATURAN KECEPATAN MOTOR INDUKSI SATU FASA DENGAN PWM MENGGUNAKAN PENGENDALI PID BERBASIS ARDUINO

Sejauh yang saya ketahui bukan merupakan hasil duplikasi dari skripsi dan/atau karya ilmiah lainnya yang pernah dipublikasikan dan/atau pernah dipergunakan untuk mendapatkan gelar kesarjanaan di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Bengkulu, Juli 2014



TIRTA WISESA

G1D009060

KATA PENGANTAR

Alhamdulillah *robbil 'alamin*, Puja dan puji syukur penulis ucapkan kepada Allah *subhanawataala* karena berkat rahmat, karunia dan segala nikmat-Nya penulis dapat menyelesaikan penulisan skripsi yang berjudul **“Perancangan Pengaturan Kecepatan Motor Induksi Satu Fasa dengan PWM Menggunakan Pengendali PID Berbasis Arduino”**

Penulisan skripsi ini dimaksudkan untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Jenjang Strata Satu (S1) pada Program Studi Teknik Elektro. Penulis menyadari bahwa dalam penulisan skripsi ini masih jauh dari kata sempurna, oleh karena itu saran dan kritik yang membangun sangat diharapkan untuk bahan pertimbangan dan perbaikan untuk karya selanjutnya.

Terselesaikannya penulisan skripsi ini tak terlepas dari sumbangsih berbagai pihak. Maka untuk segala bentuk bantuan, motivasi, bimbingan, pengarahan dan kerjasama yang baik penulis ingin mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. M. Khairul Amri Rosa, S.T., M.T. sebagai dosen pembimbing utama skripsi, yang telah banyak memberikan arahan, bimbingan serta mengajarkan tentang arti kata perjuangan dan kesabaran yang sesungguhnya.
2. Ika Novia Anggraini, S.T., M.Eng. selaku dosen pembimbing pendamping skripsi, yang terus menyemangati dan dengan sangat baik membimbing penulis dalam penyelesaian skripsi ini.
3. Irnanda Priyadi, S.T., M.T. selaku dosen penguji skripsi serta pembimbing akademik, yang telah membimbing penulis dari tahun awal perkuliahan hingga sekarang serta memberikan saran dan kritik yang sangat membangun.
4. Afriyastuti Herawati, S.T., M.T. selaku dosen penguji skripsi, beliau menginspirasi penulis tentang banyak hal luar biasa.
5. Seluruh dosen pengampuh di Program Studi Teknik Elektro yang telah membagi ilmu pada penulis.
6. Kedua Orang tua terkasih sepanjang hidup, Ayuk dan Adik - adik dan seluruh keluarga besarku yang telah menyemangatiku selama ini.

7. Untuk keluarga baruku di Laboraturium Elektro. Terima kasih, Afit, Munadi, Anggo, Raimon, Atria, Roli Koeman, Rikardo, Kak Anom, Kak Joko, Emil, Dana, Aji, Suman, Riski, Rahmat, Pandi teman seperjuangan yang selalu mendukung, menghibur dan mendengarkan setiap keluhanku selama ini.
8. Terima kasih untuk teman-teman Teknik Elektro angkatan 2009, telah melalui perkuliahan bersama selama kurang lebih 5 tahun. Terima kasih atas semua pertolongan yang kalian berikan.
9. Terima kasih untuk Piko Saputra, kak Johan, Kak Rico Bernando, Bang Ivan, Adi dan teman-teman lainnya yang telah dengan baik hati membagi ilmu dan membantu dalam penyelesaian skripsi ini.
10. Keluarga baruku di Bengkulu, Bunda dan anak - anaknya, Wina, Adam, dan semua teman yang tidak bisa disebut satu per satu. Terima kasih atas dukungan kalian semua

Penulis hanya dapat mengucapkan terimakasih dengan cara yang sederhana ini, semoga Allah membalas semua kebaikan besar yang diberikan. Akhirnya penulis berharap bahwa skripsi ini dapat bermanfaat bagi seluruh pembaca dan memberikan kontribusi terhadap perkembangan Program Studi Teknik Elektro.

Billahitaufiq Wal Hidayah

Wassalamu'alaikum Wr. Wb

Bengkulu, Juli 2014

Tirta Wisesa

Abstrak

Pengaturan kecepatan motor induksi satu fasa dalam penelitian ini menggunakan pengendali *Proportional, Integral, Derivativ* (PID) untuk mengatur kecepatan motor untuk mendapatkan kecepatan yang tetap walaupun terjadi perubahan beban. Pengaturan kecepatan ini dikendalikan oleh pulsa PWM untuk mendapatkan frekuensi yang sesuai dengan kecepatan referensi. Pada penelitian ini, pengujian tanpa beban didapat *rise time* motor akan semakin cepat pada kecepatan referensi yang semakin kecil. Pada kecepatan 2500 rpm *rise time* sebesar 2,65 s dan pada kecepatan 1800 rpm *rise time* sebesar 1,89 s. Pada pengujian berbeban didapat nilai *steady state time* motor pada kecepatan 2500 rpm yaitu pada beban naik 3,88 s dan beban turun 2,75 s. Pada kecepatan 1800 rpm yaitu pada beban naik 2,46 s dan beban turun 1,58 s. Konsumsi daya motor saat menggunakan sistem pengendali kecepatan, pengujian tanpa beban di kecepatan 2500 rpm yaitu 52,20 watt dan di kecepatan 1800 rpm yaitu 14,02 watt. Pada pengujian berbeban di kecepatan 2500 rpm yaitu 48,09 watt dan di kecepatan 1800 rpm yaitu 14,01 watt.

Kata kunci : Arduino, motor induksi satu fasa, *Proportional Integral Derivativ* (PID) , *rise time*.

DAFTAR ISI

| | |
|----------------------------------|------|
| HALAMAN JUDUL..... | i |
| LEMBAR PENGESAHAN | ii |
| LEMBAR PERNYATAAN KEASLIAN | iii |
| MOTTO DAN PERSEMBAHAN | iv |
| KATA PENGANTAR | v |
| ABSTRAK | vi |
| DAFTAR ISI..... | viii |
| DAFTAR GAMBAR | xi |
| DAFTAR TABEL..... | xiii |

BAB 1 Pendahuluan

| | |
|----------------------------|---|
| 1.1 Latar Belakang | 1 |
| 1.2 Perumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan..... | 3 |

BAB 2 Tinjauan Pustaka

| | |
|--|----|
| 2.1. Motor Induksi..... | 5 |
| 2.1.1 Kecepatan Motor Induksi | 5 |
| 2.1.2 Motor Induksi Satu Fasa | 7 |
| 2.1.3 Pengaturan Kecepatan Motor Induksi | 9 |
| 2.2 Konverter Daya | 10 |
| 2.2.1 Penyearah | 10 |
| 2.2.2 Inverter <i>Voltage-source</i> Satu Fasa | 12 |
| 2.2.3 <i>Pulse Width Modulation</i> (PWM) | 13 |
| 2.3 LCD (<i>Liquid Crystal Display</i>) | 15 |
| 2.4 Sensor Pendeteksi Kecepatan..... | 16 |
| 2.5 Mikrokontroler Berbasis Arduino | 16 |
| 2.5.1 Lingkungan Pemrograman Arduino | 17 |
| 2.5.2 Struktur program Arduino | 18 |
| 2.5.3 Variabel dan Tipe Data Arduino | 20 |
| 2.5.4 Fungsi | 20 |
| 2.5.5 <i>Board</i> Arduino uno | 21 |
| 2.6 Pengendali PID | 23 |
| 2.6.1 Kontrol Proporsional..... | 24 |

...

| | |
|--|----|
| 2.6.2 Kontrol Integral | 24 |
| 2.6.3 Kontrol Derivatif | 24 |
| 2.6.4 Persamaan PID | 25 |
| 2.7 Metode Ziegler Nichols & <i>Good Gain Method</i> | 25 |
| 2.7.1 Karakteristik Proporsional | 26 |
| 2.7.2 Karakteristik Integral | 26 |
| 2.7.3 Karakteristik Derivatif | 26 |
| 2.7.4 Ziegler-Nichols <i>Step Response Method</i> | 26 |
| 2.7.5 Ziegler-Nichols <i>Frequency Response Method</i> | 27 |
| 2.7.6 <i>Good Gain Method</i> | 28 |

BAB 3 Metode Penelitian

| | |
|---------------------------------------|----|
| 3.1 Waktu Pengujian | 30 |
| 3.2 Objek Penelitian | 30 |
| 3.3 Metode Perancangan | 30 |
| 3.3.1 Alat dan Bahan | 31 |
| 3.3.2 Tahapan Penelitian | 32 |
| 3.4 Perancangan <i>Hardware</i> | 32 |
| 3.4.1 Arduino | 34 |
| 3.4.2 Inverter Daya | 34 |
| 3.4.3 <i>Optocoupler</i> | 35 |
| 3.4.4 LCD | 35 |
| 3.5 Perancangan <i>Software</i> | 35 |
| 3.5.1 Perancangan PID | 35 |
| 3.6 Analisis kerja | 36 |
| 3.7 Pengujian | 36 |
| 3.8 Diagram Alir Sistem Kendali | 37 |

BAB 4 Hasil Dan Pembahasan

| | |
|---|----|
| 4.1 Pengujian <i>Rise Time</i> Motor | 39 |
| 4.1.1 Pengujian <i>Rise Time</i> tanpa beban | 39 |
| 4.1.2 Perbandingan Optimasi <i>Rise time</i> motor | 40 |
| 4.1.2 Pengujian <i>Steady State Time</i> Motor Berbeban | 41 |

| | |
|--|----|
| 4.2 Pengujian Efisiensi Daya | 46 |
| 4.2.1 Pengujian Efisiensi Daya Tanpa Beban | 46 |
| 4.2.2 Pengujian Efisiensi Daya Berbeban | 48 |
| 4.2.3 Pengukuran dan Perhitungan Efisiensi daya tanpa beban..... | 49 |
| 4.2.4 Pengukuran dan Perhitungan Efisiensi daya berbeban | 51 |

BAB 5 Penutup

| | |
|----------------------|----|
| 5.1 Kesimpulan | 53 |
| 5.2 Saran | 53 |

Daftar Pustaka

Lampiran

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Rangkaian Ekivalen & Rangkaian Pengganti Motor | 8 |
| Gambar 2.2 Kontruksi Motor Induksi | 8 |
| Gambar 2.3 Motor induksi <i>Split phase</i> & arus motor | 9 |
| Gambar 2.4 Karakteristik Pengaturan Kecepatan | 10 |
| Gambar 2.5 Rangkaian Penyearah Gelombang Penuh | 11 |
| Gambar 2.6 Bentuk Gelombang <i>Output</i> dan <i>Input</i> Penyearah..... | 11 |
| Gambar 2.7 Bentuk Gelombang <i>output</i> penyearah | 12 |
| Gambar 2.8 Inverter satu fasa jembatan penuh | 13 |
| Gambar 2.9 VSI jembatan penuh dengan SPWM..... | 14 |
| Gambar 2.10 Rangkaian dasar system konverter | 15 |
| Gambar 2.11 LCD 16x2..... | 15 |
| Gambar 2.12 rangkaian Optocoupler | 16 |
| Gambar 2.13 Tampilan IDE Arduino | 18 |
| Gambar 2.14 Board Arduino Uno | 22 |
| Gambar 2.15 Blok diagram pengendali PID | 23 |
| Gambar 2.16 Diagram Blok Proporsional..... | 24 |
| Gambar 2.17 Diagram Blok Integral..... | 24 |
| Gambar 2.18 Diagram Blok Derifatif | 25 |
| Gambar 2.19 Kurva Respon Langkah | 27 |
| Gambar 2.20 Karaktristik keluaran dengan penambahan Kp | 28 |
| Gambar 2.21 <i>Ultimate Gain</i> | 28 |
| Gambar 2.22 Respon T_{ou} | 28 |
| Gambar 3.1 Diagram Blok Sistem | 33 |
| Gambar 3.2 Pengkodean PID | 34 |
| Gambar 3.3 Diagram Alir Sistem Pengendali | 37 |
| Gambar 4.1 Hasil Perancangan Alat | 38 |
| Gambar 4.2 Grafik Perbandingan <i>Rise Time</i> | 40 |
| Gambar 4.3 Grafik Perbandingan <i>Steady State</i> | 41 |
| Gambar 4.4 Grafik Perbandingan <i>Steady State</i> | 42 |
| Gambar 4.5 Grafik Perbandingan <i>Steady State</i> | 43 |

| | |
|--|----|
| Gambar 4.6 Grafik Perbandingan <i>Steady State</i> | 44 |
| Gambar 4.7 Grafik Perbandingan <i>Steady State</i> | 45 |
| Gambar 4.8 Pengujian Daya <i>Output</i> dan <i>Input</i> | 47 |
| Gambar 4.9 Pengujian Daya <i>Output</i> dan <i>Input</i> | 48 |
| Gambar 4.10 Perbandingan Daya <i>Input</i> Ukur dan Hitung tanpa beban..... | 48 |
| Gambar 4.11 Perbandingan Daya <i>output</i> Ukur dan Hitung tanpa beban..... | 49 |
| Gambar 4.12 Perbandingan Daya <i>Input</i> Ukur dan Hitung berbeban | 50 |
| Gambar 4.13 Perbandingan Daya <i>output</i> Ukur dan Hitung berbeban | 50 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Fungsi IDE Arduino..... | 19 |
| Tabel 2.2 Keterangan data Variabel..... | 20 |
| Tabel 2.3 Spesifikasi Arduino Uno..... | 22 |
| Tabel 2.4 Parameter PID Kurva Respon Langkah..... | 27 |
| Tabel 3.1 Data Spesifikasi Motor | 30 |
| Tabel 4.1 Nilai Kp, Ki & Kd..... | 39 |
| Tabel 4.2 Pengujian <i>Rise Time</i> tanpa Beban..... | 39 |
| Tabel 4.3 Perbandingan <i>Rise Time</i> motor induksi 3 fasa dan 1 fasa..... | 40 |
| Tabel 4.4 pengujian <i>Steady state</i> kecepatan referensi 2500 Rpm..... | 41 |
| Tabel 4.5 pengujian <i>Steady state</i> kecepatan referensi 2300 Rpm..... | 42 |
| Tabel 4.6 pengujian <i>Steady state</i> kecepatan referensi 2200 Rpm..... | 43 |
| Tabel 4.7 pengujian <i>Steady state</i> kecepatan referensi 1900 Rpm..... | 44 |
| Tabel 4.8 pengujian <i>Steady state</i> kecepatan referensi 1800 Rpm..... | 45 |
| Tabel 4.9 Data Pengujian Efisiensi Daya tanpa Beban | 46 |
| Tabel 4.10 Data Pengujian Efisiensi Daya Berbeban..... | 47 |
| Tabel 4.11 Daya pengukuran dan perhitungan tanpa beban | 48 |
| Tabel 4.12 Daya pengukuran dan perhitungan Berbeban | 50 |

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Motor induksi satu fasa digunakan secara luas pada berbagai aplikasi komersil dan rumah tangga karena harga yang murah, handal dan kuat. Sumber tegangan tiga fasa umumnya tidak tersedia pada pelanggan rumah tangga, oleh karena itu motor induksi satu fasa merupakan pilihan utama karena dapat langsung dihubungkan dengan sumber tegangan AC.

Kecepatan putaran motor induksi sangat dipengaruhi oleh frekuensi tegangan sumber dan beban mekanisnya. Pada keadaan tanpa beban, kecepatan putaran motor induksi mendekati kecepatan medan magnet putar (kecepatan sinkron). Bila motor dibebani, kecepatan putaran motor berkurang akibat adanya torka beban yang menyebabkan slip pada motor induksi.

Dalam pengoperasian motor induksi satu fasa, kecepatan putaran motor tidak konstan. Kecepatan maksimum motor terjadi ketika motor tidak dibebani. Kecepatan motor jauh berkurang karena adanya beban yang harus dipikul oleh motor. Hal ini mengindikasikan bahwa kecepatan putaran diperlukan adalah kecepatan dimana motor sedang mendapat beban penuh.

Rentang waktu dimana motor berputar tanpa beban cukup signifikan. Jika kecepatan motor pada waktu tersebut dapat diturunkan, konsumsi energi listrik menjadi berkurang. Konsekuensinya biaya yang harus dibayarkan untuk konsumsi energi listrik PLN bisa menjadi lebih murah. Pengaturan kecepatan dapat dilakukan dengan mengubah frekuensi sumber tegangan listrik. Untuk melakukannya diperlukan sebuah konverter daya yang terdiri atas penyearah tegangan listrik dan inverter beserta sistem kontrol yang sesuai.

Konverter daya yang akan digunakan adalah konverter daya yang terdiri atas penyearah gelombang penuh yang akan menghasilkan tegangan DC. Tegangan DC yang dihasilkan oleh penyearah difilter menggunakan kapasitor untuk mendapatkan tegangan DC yang lebih murni atau tegangan DC yang lebih halus karena *ripple* yang dihasilkan akan mengecil akibat pemasangan kapasitor tersebut. Tegangan DC diubah kembali menjadi tegangan AC menggunakan inverter. Tegangan bolak-

balik yang dihasilkan sebagai input motor induksi menggunakan inverter satu fasa jembatan penuh menggunakan sakelar MOSFET (*Metal Oxide Silicon Field Effect Transisto*). Rangkaian penyearah dan inverter menggunakan jembatan penuh karena mampu menghasilkan transfer daya yang efisien dan penggunaan tegangan yang lebih optimal dibanding rangkaian setengah jembatan. Metode yang digunakan untuk memodulasi inverter adalah *sinusoidal pulse-width modulation* (SPWM) unipolar.

Untuk mendapatkan kecepatan putaran yang diharapkan digunakan sistem kontrol dengan pengendali PID. Pengendali PID mudah untuk diterapkan dan berbiaya rendah. Pengendali PID diterapkan pada sistem Arduino pada mikrokontrolernya. Arduino merupakan sistem yang lebih kompleks dalam pengaturan kecepatan motor. Photodiode digunakan sebagai sensor untuk membaca kecepatan putaran motor. Kecepatan referensi dibandingkan dengan kecepatan yang terukur diumpan kepada pengendali sebagai sinyal *error*. Pengendali mengambil keputusan berdasarkan *error* yang terbaca dengan mengirimkan sinyal kepada inverter SPWM untuk membangkitkan frekuensi yang tepat. Frekuensi yang dihasilkan akan menjaga kecepatan motor induksi pada posisi kecepatan referensinya walaupun beban yang diberikan pada motor induksi satu fasa berubah – ubah.

Penelitian ini akan diimplementasikan pada motor induksi satu fasa. Dimana, motor ini akan diatur kecepatannya pada kecepatan saat berbeban dan kecepatan tersebut tidak berubah-ubah walaupun motor dalam keadaan berbeban dan tidak berbeban.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, perumusan masalah penelitian antara lain:

1. Bagaimana kecepatan putaran mesin pada keadaan tanpa beban diturunkan sama dengan kecepatan berbeban. Menjaga kecepatan motor konstan baik pada keadaan tanpa beban, berbeban dan perubahan beban.
2. Bagaimana merancang rangkaian penyearah dan inverter frekuensi variabel untuk menghasilkan variasi kecepatan sinkron.

3. Bagaimana desain sistem kendali PID untuk motor induksi agar dapat menghasilkan respon yang cepat untuk mengatasi perubahan beban.

1.3 Batasan Masalah

1. Kecepatan putaran motor hanya diatur oleh frekuensi yang dibangkitkan.
2. Hanya satu kecepatan referensi yang digunakan pada beban yang berbeda.
3. Tidak membahas tentang filter yang digunakan.
4. Hanya pengendali PID (*Proportional, Integral, Derivativ*) yang digunakan untuk menentukan nilai set poin.
5. Inverter yang digunakan adalah inverter terkendali berbasis PWM (*Pulse Width Modulation*).
6. Sistem mikrokontroler yang digunakan adalah Arduino.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Membuat *prototype* sistem pengontrolan kecepatan pada motor induksi satu fasa.
2. Merancang perangkat lunak pendeteksi kecepatan dan pengaturan besar frekuensi yang dibangkitkan menggunakan Arduino dengan pengendali PID.
3. Merancang sistem pengaturan kecepatan yang stabil pada kecepatan referensi yang telah ditentukan.
4. Analisa konsumsi daya menggunakan sistem pengendali kecepatan pada motor induksi satu fasa.

BAB 2

TINJAUAN PUSTAKA

Pada tahun 2010 telah dilakukan penelitian berjudul kontrol PID untuk pengaturan kecepatan motor pada *prototype* ayunan bayi otomatis. Penelitian ini bertujuan membuat sistem sebuah ayunan bayi otomatis yang mampu bergerak setelah mendeteksi suara tangisan bayi dengan gerakan perlahan dan stabil setelah beberapa jam. Pada penelitian ini metode pengendali yang digunakan adalah menggunakan PID (*Proportional, Integral, Derivativ*) dikarenakan PID mampu menghasilkan gerakan pengaturan kecepatan yang lebih halus dan stabil. Penelitian ini berdasarkan metode tuning PID Ziegler Nichols yang mana metode ini menggunakan pengambilan garis singgung dengan cara membuat garis lurus dan menghimpit pada respon yang memiliki nilai linear paling banyak dan melalui nilai torka yaitu waktu yang diperlukan oleh respon untuk mencapai 63,2% dari nilai *stedy state respon*. Penelitian ini juga menggunakan fungsi matlab untuk menentukan dan mensimulasikan *transfer function* apakah antara pengukuran dan simulasi telah didapat kesamaan dalam mencapai keadaan *stedy state* pada motor [1].

Pada tahun 2011 dilakukan penelitian tentang pengaturan kecepatan motor yang berjudul pembuatan modul inverter sebagai kendali kecepatan motor induksi. Pada penelitian ini kendali V/f konstan adalah salah satu cara untuk mengendalikan kecepatan putar motor induksi dengan mengubah tegangan dan frekuensi. Hal yang paling umum dalam penerapan cara ini adalah dengan menggunakan perangkat yang dikenal sebagai inverter. Oleh karena itu, pada penelitian ini penulis merancang inverter, khususnya inverter satu fasa dengan kendali V/f konstan, yang diaplikasikan untuk mengendalikan kecepatan putar motor induksi. Berdasarkan hasil pengujian, nilai rata-rata konstanta rasio V/f yang didapat yaitu 2,34. Inverter yang telah dirancang mampu mengatur kecepatan putar motor induksi dengan baik, rentang pengaturan yang dapat dicapai sangat lebar yaitu dari 262 rpm dengan frekuensi 10 Hz sampai dengan 1826 rpm pada frekuensi 60 Hz pada keadaan tanpa beban dengan kenaikan dan penurunan setiap 1 Hz. Sedangkan perubahan motor dapat dengan halus, rata-rata 31,2 rpm/Hz [2].

Penelitian selanjutnya berjudul pengaruh injeksi harmonisa orde ketiga terhadap kecepatan motor induksi dengan kendali volt/hertz. Tujuan dari pembuatan penelitian ini adalah mengendalikan kecepatan motor induksi dengan mengubah frekuensi dari 50 Hz hingga 150 Hz. Dalam penelitian ini dibahas tentang pengaruh injeksi harmonisa orde ketiga terhadap kinerja motor induksi tiga fasa sebagai alat penggerak berbasis kendali volt/hertz. Suatu uji coba dilakukan dengan menggunakan sinyal 50 Hz kemudian dilanjutkan dengan sistem injeksi harmonisa orde ketiga yaitu 150 Hz dengan kombinasi amplitude. Dari hasil uji coba skala laboratorium alat ini dapat bekerja dengan baik dengan frekuensi operasi 5 sampai 50 Hz dan menunjukkan operasi paling baik pada amplituda 0.25 [3].

Penelitian yang akan dilakukan memiliki perbedaan dengan penelitian-penelitian sebelumnya seperti pada penelitian ini dilakukan pada alat mesin motor induksi serta dilakukan menambahkan sistem minimum mikrokontroler terbaru yaitu Arduino dan pengendali PID untuk mengontrol lebar pulsa untuk mendapatkan frekuensi yang diinginkan agar didapatkan kecepatan motor induksi satu fasa yang sesuai dengan kecepatan referensi atau kecepatan yang diinginkan.

2.1. Motor Induksi

Motor induksi merupakan jenis motor elektrik yang paling luas dipergunakan untuk berbagai aplikasi dalam industri, disebabkan harga yang relatif murah dan memiliki keandalan yang tinggi serta memiliki range daya yang cukup luas untuk digunakan kepada beban yang besar. Motor induksi memiliki konstruksi yang sederhana, praktis, dan hampir bebas biaya perawatan. Kontruksi motor induksi terbagi atas rotor dan stator. Motor induksi satu fasa dapat secara langsung dioperasikan dengan menghubungkannya kepada sumber tegangan listrik di rumah atau gedung. Saat ini motor induksi telah menggantikan banyak peran motor DC.

2.1.1. Kecepatan motor induksi

Kecepatan putaran motor induksi ditentukan oleh kecepatan medan magnet putar (sinkron) dan slip akibat beban yang dihubungkan dengan poros motor. Kecepatan sinkron terutama ditentukan oleh frekuensi sumber tegangan listrik.

$$n_s = \frac{120f}{P} \quad (2.1)$$

dengan,

n_s = kecepatan sinkron, rpm.

f = frekuensi jala-jala, Hz.

P = jumlah kutub motor.

Kecepatan putaran rotor selalu lebih rendah dari pada kecepatan sinkron. Perbedaan relatif antara putaran rotor dan kecepatan sinkron disebut slip (S) [4]. Nilai slip dipengaruhi oleh beban. Pada keadaan tanpa beban, slip motor induksi sangat rendah sehingga putaran rotor mendekati kecepatan sinkron. Jika motor dibebani maka slip akan bertambah sehingga putaran motor menjadi berkurang.

$$S = \frac{n_s - n_r}{n_s} \times 100\% \quad (2.2)$$

$$n_r = (1 - S)n_s \quad (2.3)$$

dengan,

S = slip

n_r = kecepatan putaran rotor, rpm.

Persamaan daya sumber dan persamaan daya pada motor induksi satu fasa dinyatakan dengan:

$$P = VI \quad (2.4)$$

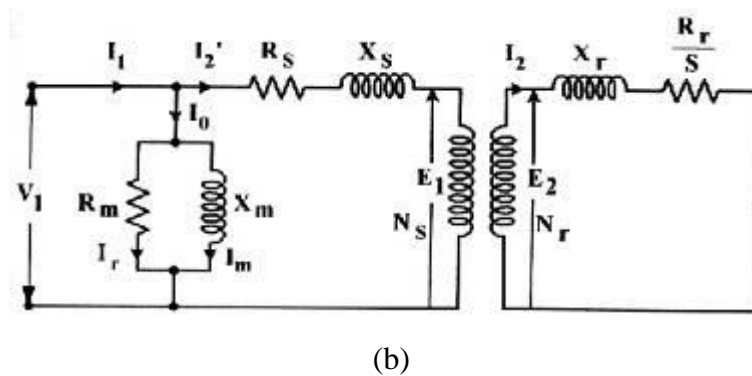
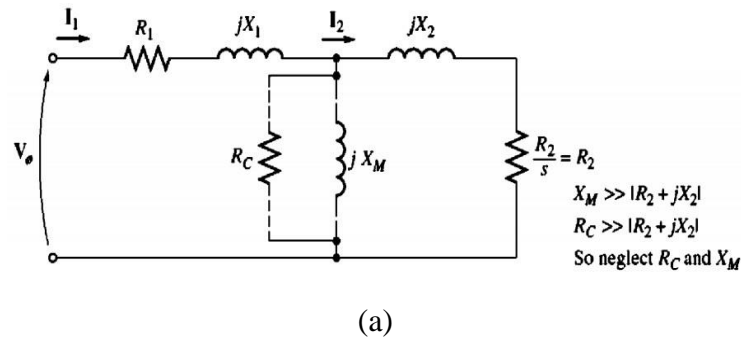
$$P = V_T I_L \cos \theta \quad (2.5)$$

$$\eta = \text{Daya Output} / \text{Daya Input} \times 100 \quad (2.6)$$

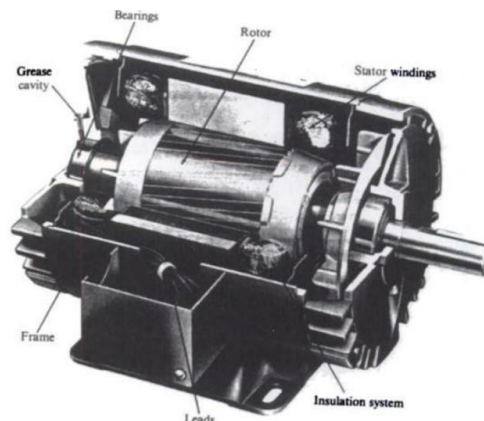
Karena pada frekuensi rendah tegangan harus diturunkan, sehingga dengan menurunkan kecepatan motor induksi, daya yang dikonsumsi menjadi berkurang.

2.1.2. Motor Induksi Satu Fasa

Motor induksi satu fasa menggunakan jenis rotor sangkar. Rangkaian ekivalen dan konstruksi motor induksi satu fasa diperlihatkan pada Gambar 2.1 [4] dan 2.2 [5].



Gambar 2.1 (a) rangkaian ekivalen motor induksi 1 fasa (b) Rangkaian pengganti motor induksi satu fasa



Gambar 2.2. Konstruksi motor induksi satu fasa dengan rotor sangkar.

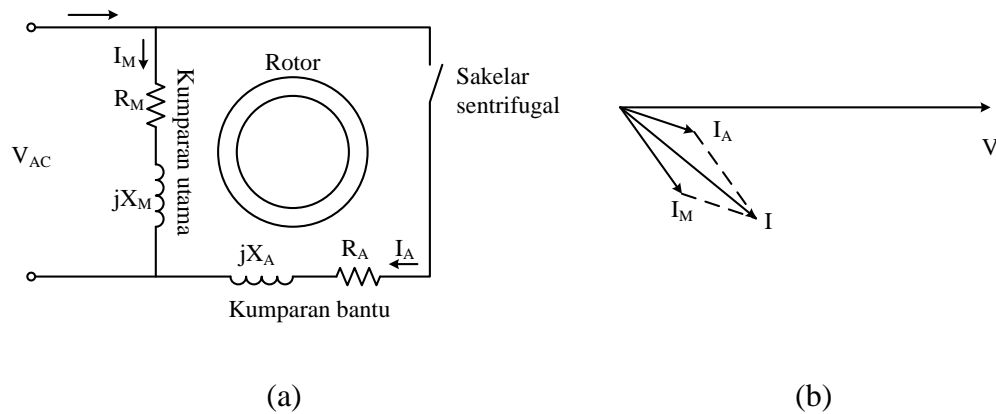
Karena hanya ada satu fasa pada kumparan stator, medan magnet yang dibangkitkan tidak berputar sehingga tidak ada tegangan yang terinduksi pada konduktor rotor. Dengan demikian motor satu fasa tidak mempunyai torka start.

Untuk menghasilkan medan magnet putar pada stator terdapat tiga teknik umum, yaitu:

1. *Split-phase*
2. *Capacitor start* dan *capacitor run*
3. *Shaded poles*.

Pada bagian ini hanya akan dibahas motor induksi satu fasa jenis *split-phase*, karena jenis motor ini yang paling banyak digunakan pada aplikasi praktis.

Untuk menghasilkan medan magnet putar, pada stator terdapat dua buah kumparan yaitu kumparan utama dan kumparan bantu yang diatur berbeda fasa sebesar 90° elektrik [4]. Rangkaian kumparan motor induksi satu fasa *split-phase* diperlihatkan Gambar 2.3.



Gambar 2.3. (a) Motor induksi *split-phase*. (b) Arus motor pada kondisi start.

Saklar sentrifugal berfungsi untuk melepas kumparan bantu setelah motor mencapai kecepatan tertentu. Kumparan bantu dirancang untuk mempunyai rasio tahanan-reaktansi yang lebih besar daripada kumparan utama dengan menggunakan bahan penghantar yang penampangnya lebih kecil. Arus pada kumparan bantu mendahului arus kumparan utama menyebabkan salah satu dari dua medan magnet stator yang berputar berlawanan lebih besar daripada yang lain.

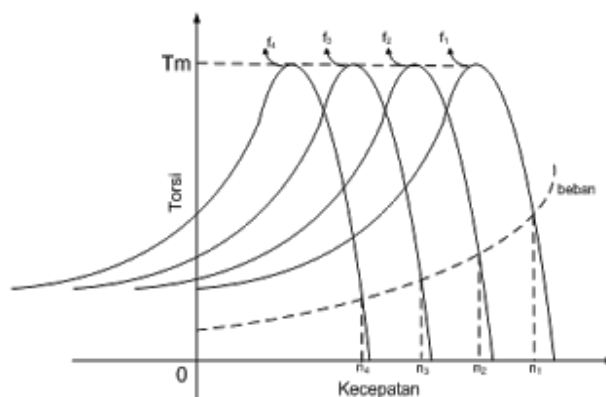
Motor induksi *split-phase* mempunyai torka start yang cukup tinggi dengan arus start yang relatif kecil [4]. Motor jenis ini cocok digunakan untuk aplikasi yang tidak memerlukan torka start yang sangat besar seperti kipas angin, *blower*, dan pompa sentrifugal. Arus dan medan magnet pada kumparan bantu selalu mencapai nilai puncak sebelum arus pada kumparan utama. Arah putaran motor dapat dibalik dengan menukar hubungan kumparan bantu.

2.1.3. Pengaturan kecepatan Motor Induksi

Pengaturan kecepatan motor induksi satu fasa rotor sangkar dapat dilakukan dengan:

1. Variasi frekuensi stator

Pengaturan putaran motor induksi dapat dilakukan dengan mengatur besar frekuensi yang dimiliki oleh motor induksi tersebut. Frekuensi berbanding lurus dengan kecepatan motor yang akan dihasilkan. Frekuensi juga berkaitan dengan torka akibat adanya beban yang diperlihatkan pada Gambar 2.4.



Gambar 2.4 karakteristik pengaturan kecepatan terhadap perubahan frekuensi

2. Perubahan jumlah kutub

Kecepatan motor induksi dapat pula diatur dengan mengubah jumlah kutubnya. Dimana, kecepatan motor induksi akan lebih cepat dengan menggunakan kutub yang lebih sedikit dibandingkan dengan ber kutub banyak.

3. Mengubah tegangan terminal V_T dan variabel arus.

Selain menggunakan frekuensi dan mengubah jumlah kutub, kecepatan motor induksi dapat juga diatur dengan mengubah tegangan terminal, semakin tegangan yang diberikan besar maka semakin besar pula kecepatan yang dihasilkan oleh motor. Perubahan tegangan terminal dan arus berhubungan dengan frekuensi, (V-f) control dan (I-f) control [6].

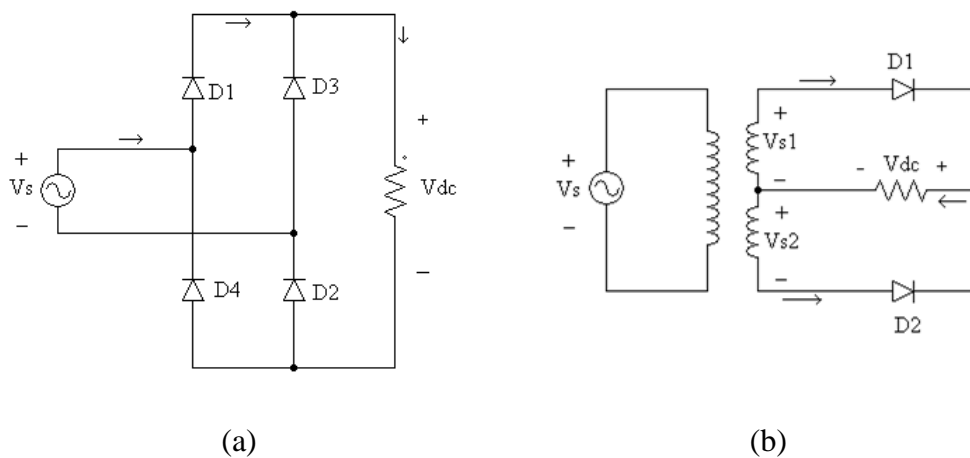
2.2. Konverter Daya

Rangkaian elektronika daya berfungsi untuk mengatur energi listrik dari sekumpulan sumber daya dan beban. Fungsi konverter daya terletak di tengah-tengah dari suatu sistem yang mengendalikan aliran energi antara sumber listrik yang tersedia dan beban yang akan dicatu. Konverter daya yang menghubungkan sumber daya dan beban sangat mempengaruhi keandalan sistem.

Pada sistem pengemudian motor induksi, konverter daya yang digunakan terdiri dari bagian yang mengonversi tegangan arus bolak balik menjadi tegangan arus searah dan bagian yang memodulasi tegangan arus searah itu menjadi arus bolak balik dengan magnitude dan frekuensi yang sesuai.

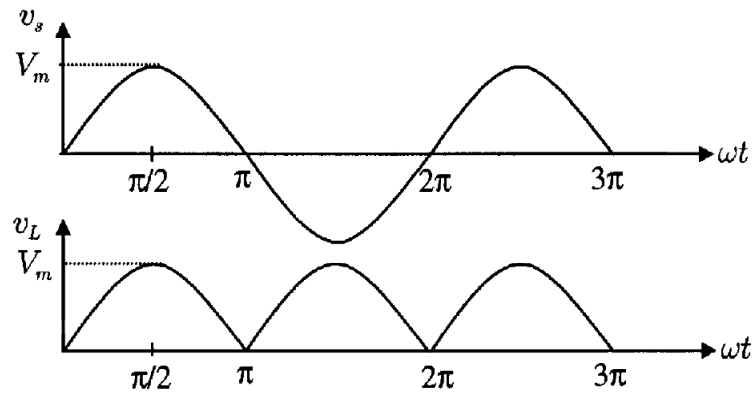
2.2.1. Penyearah

Dioda merupakan devais semikonduktor yang dapat digunakan dalam rangkaian penyearah. Rangkaian penyearah gelombang penuh dengan dioda diperlihatkan pada Gambar 2.5.



Gambar 2.5. Rangkaian penyearah gelombang penuh menggunakan dioda
(a) rangkaian jembatan penuh, (b) rangkaian penyearah dengan center tap

Bentuk gelombang *input* dan *output* rangkaian diperlihatkan pada Gambar 2.6.



Gambar 2.6. Bentuk gelombang *output* dan *input* rangkaian penyearah satu fasa jembatan penuh

Tegangan dan arus rata-rata DC yang dihasilkan penyearah gelombang penuh:

$$V_{dc} = \frac{1}{\pi} \int_0^{\pi} V_m \sin(\omega t) d\omega t = \frac{2V_m}{\pi} = 0.637V_m \quad (2.7)$$

$$I_{dc} = \frac{V_{dc}}{R} \quad (2.8)$$

Harga rms pada tegangan pada beban:

$$V_L = \sqrt{\frac{1}{\pi} \int_0^{\pi} (V_m \sin \omega t)^2 d\omega t} = \frac{V_m}{\sqrt{2}} = 0.707V_m \quad (2.9)$$

$$I_L = \frac{V_L}{R} \quad (2.10)$$

Daya keluaran rerata (P_{dc}) dan daya rms (P_L):

$$P_{dc} = V_{dc} \cdot I_{dc} \quad (2.11)$$

$$P_L = V_L \cdot I_L \quad (2.12)$$

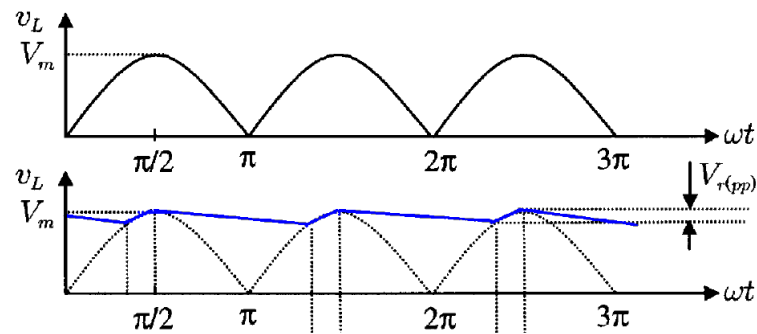
Dengan:

V_{dc} = tegangan dc

I_{dc} = arus dc

V_m = tegangan rata – rata

Untuk memperbaiki bentuk gelombang output penyearah agar ripple yang dihasilkan menjadi berkurang, pada terminal *output* dioda ditambahkan filter kapasitif. Ketika nilai sesaat v_s lebih tinggi daripada tegangan kapasitor, energi diberikan untuk mengisi kapasitor. Kemudian, pada saat tegangan v_s turun di bawah tegangan kapasitor, muatan yang tersimpan pada kapasitor dilepas kepada tahanan beban R . Bentuk gelombang output penyearah dengan filter kapasitor diperlihatkan pada Gambar 2.7.



Gambar 2.7. Bentuk gelombang *output* penyearah sebelum dan setelah pemasangan filter kapasitif.

Persamaan tegangan setelah pemasangan filter kapasitif adalah:

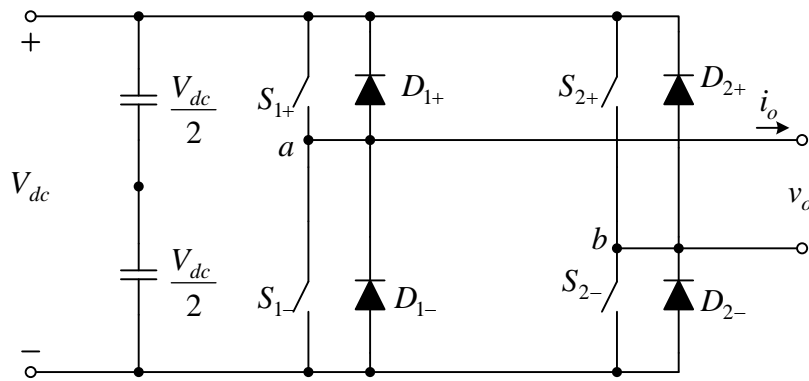
$$V_{dc} = V_m \left(1 - \frac{1}{2f_r RC} \right) \quad (2.13)$$

$$V_s = \frac{V_m}{2\sqrt{2}f_r RC} \quad (2.14)$$

2.2.2. Inverter *voltage-source* satu fasa

Inverter *voltage-source* (VSI) berfungsi untuk mengubah tegangan arus searah menjadi tegangan bolak-balik satu fasa [6]. Inverter satu fasa dapat dijumpai dalam topologi setengah jembatan dan jembatan penuh. Inverter jenis ini banyak digunakan pada perangkat *power supply* dan UPS satu fasa.

Rangkaian umum inverter satu fasa jembatan penuh diperlihatkan pada Gambar 2.8.

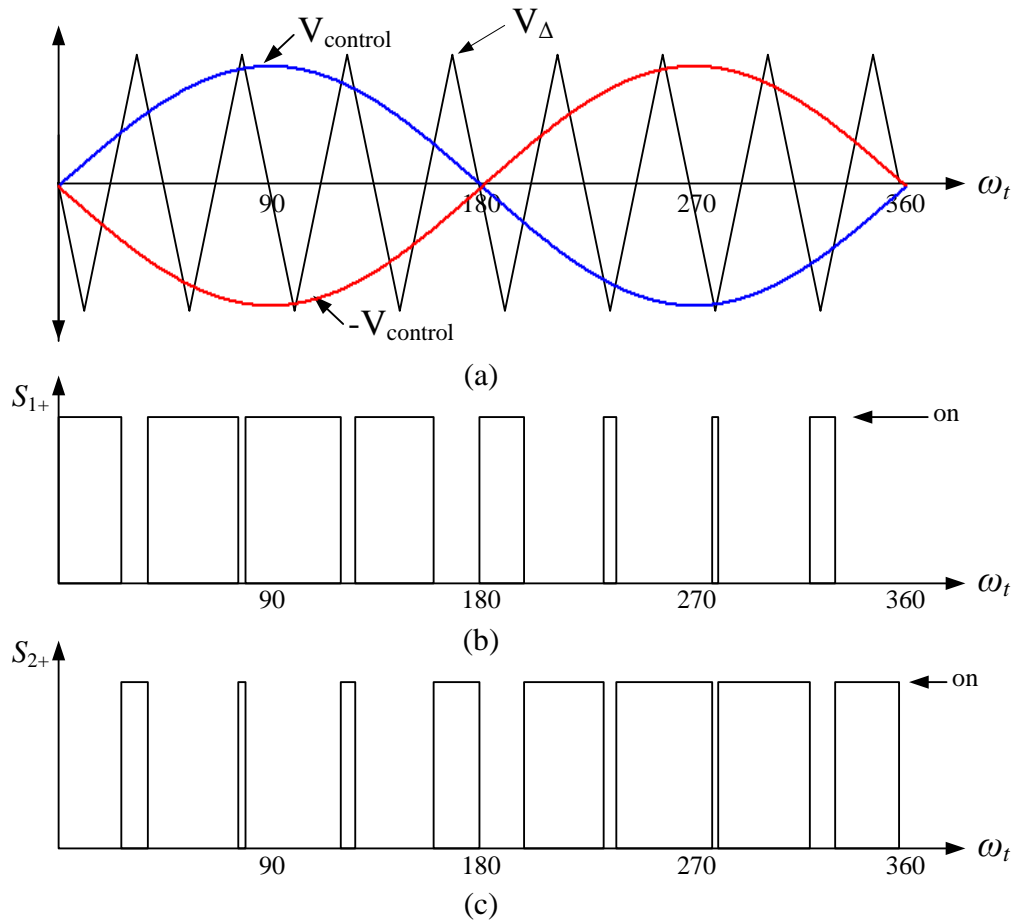


Gambar 2.8. Inverter satu fasa jembatan penuh

Inverter tiga fasa jembatan penuh terdiri atas dua kaki inverter. Susunan ini menghasilkan rating daya yang lebih tinggi daripada inverter setengah jembatan. Dengan tegangan *input* DC yang sama, tegangan *output* maksimum yang dihasilkan dua kali tegangan *output* inverter setengah jembatan. Hal ini juga berlaku untuk dayanya, sedangkan arus *output* dan arus pensakelaran adalah setengah dari inverter setengah jembatan.

2.2.3. Pulse Width Modulation (PWM)

Teknik modulasi inverter jembatan penuh dapat dilakukan menggunakan PWM unipolar. Pada teknik ini kontrol masing-masing kaki inverter dilakukan secara terpisah dengan membandingkan gelombang segitiga dengan dua gelombang sinusoidal seperti diperlihatkan pada Gambar 2.9 [7].



Gambar 2.9. VSI jembatan penuh dengan teknik SPWM unipolar.

(a) sinyal pembawa dan sinyal kontrol, (b) state sakelar 1+, (c) state sakelar 2+

Perbandingan V_{control} dengan gelombang segitiga menghasilkan sinyal untuk mengontrol sakelar pada kaki 1 sebagai berikut:

$$V_{\text{control}} > V_{\Delta} \quad : \quad T_{1+} \text{ ON} \quad \text{dan} \quad v_{AN} = V_{\text{dc}}$$

$$V_{\text{control}} < V_{\Delta} \quad : \quad T_{1-} \text{ ON} \quad \text{dan} \quad v_{AN} = 0$$

Untuk kontrol sakelar-sakelar pada kaki 2, $-V_{\text{control}}$ dibandingkan dengan gelombang segitiga.

$$(-V_{\text{control}}) > V_{\Delta} \quad : \quad T_{2+} \text{ ON} \quad \text{dan} \quad v_{BN} = V_{\text{dc}}$$

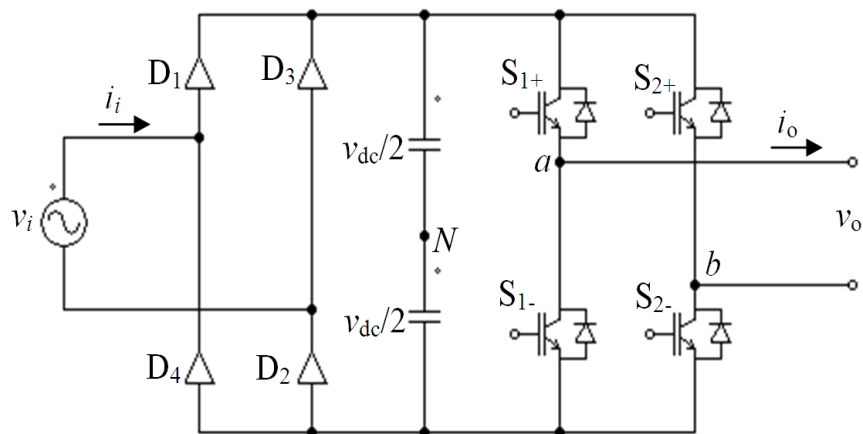
$$(-V_{\text{control}}) < V_{\Delta} \quad : \quad T_{2-} \text{ ON} \quad \text{dan} \quad v_{BN} = 0$$

Keempat sakelar membentuk empat kombinasi keadaan ON dan tegangan yang dihasilkannya sebagai berikut:

1. T_{1+}, T_{2-} ON : $v_{AN} = V_{dc}, v_{BN} = 0, v_o = V_d$
2. T_{1-}, T_{2+} ON : $v_{AN} = 0, v_{BN} = V_{dc}, v_o = -V_d$
3. T_{1+}, T_{2+} ON : $v_{AN} = V_{dc}, v_{BN} = V_{dc}, v_o = 0$
4. T_{1-}, T_{2-} ON : $v_{AN} = 0, v_{BN} = 0, v_o = 0$

Untuk menghasilkan variasi frekuensi *output* inverter, frekuensi sinyal kontrol sinusoidal disesuaikan dengan frekuensi yang dibutuhkan. Untuk mengendalikan gelombang sinusoidal tersebut diperlukan suatu pengendali. Pengendali berfungsi untuk membandingkan nilai referensi dengan nilai aktual.

Gambar 2.10 memperlihatkan sistem gabungan penyearah dan inverter satu fasa untuk mengemudikan sebuah motor induksi [7].



Gambar 2.10. Rangkaian dasar sistem konverter daya satu fasa gelombang penuh

2.3 LCD (*Liquid Crystal Display*) 16x2

LCD merupakan suatu sistem yang dapat menampilkan data dalam bentuk karakter. Data yang ditampilkan pada LCD merupakan data ASCII (*American Standard Code Internasional Interchange*), data ini telah mengkodekan nilai alphabet dan numerik menjadi data digital. LCD tipe 16x2 ini memiliki 2 baris dimana masing – masing baris memuat 16 karakter. Bentuk fisik dari LCD 16x2 diperlihatkan pada gambar 2.11 [8].

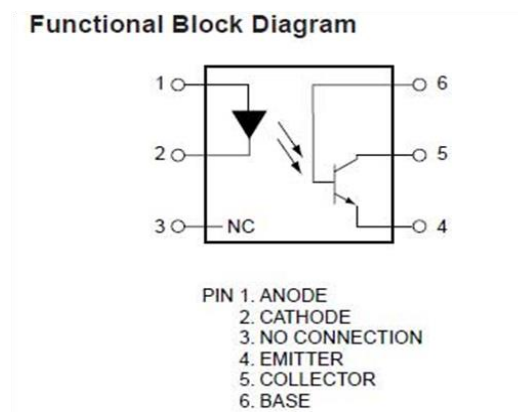


Gambar 2.11 LCD 16x2

2.4 Sensor Pendeteksi Kecepatan

Sensor pendeteksi kecepatan yang digunakan adalah sensor *optocoupler*. Sensor ini menggunakan gabungan fotodioda dan fototransistor. Fotodioda sebagai sumber cahaya dipasang langsung dengan sumber tegangan. Keluaran dari sumber cahaya akan berbanding lurus dengan tegangan masukan pada fotodioda. *Optocoupler* merupakan paket elektronik murni, jalur cahaya didalamnya yakni inframerah tertutup dalam sebuah paket. Ini menyebabkan transfer energi listrik dalam satu arah, dari inframerah ke fotodetektor, sambil mempertahankan isolasi listrik. Fungsi *optocoupler* pada umumnya selain sebagai sensor (dengan kemasan tertentu) digunakan pula pada rangkaian listrik sebagai isolasi dari rangkaian kendali dan rangkaian tegangan tinggi (daya). *Optocoupler* digunakan untuk mengatasi perpindahan informasi dari tegangan tinggi ke tegangan rendah, atau sebaliknya. Pada rangkaian *optocoupler* terjadi isolasi tegangan tinggi antara sinyal masukan dan keluaran [8]. Adapun cara kerja *optocoupler* adalah sebagai berikut:

1. *Optocoupler* bekerja berdasarkan pancaran cahaya LED yang mengenai penerima peka cahaya, misalnya fotodioda dan fototransistor.
2. Sinyal dioda diberikan ke titik 1 – 2 dan sinyal keluaran diperoleh dari titik 4 – 5. Disini terjadi isolasi listrik antara masukan dan keluaran, karena informasi disampaikan melalui cahaya [7]. Diperlihatkan pada Gambar 2.12.



Gambar 2.12 Rangkaian *optocoupler*

2.5 Mikrokontroler Berbasisi Arduino

Dalam buku “*Getting Started with Arduino*”, Arduino dituliskan sebagai sebuah *platform* komputasi fisik (*physical computing*) yang *open source* pada

board input output sederhana. Yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespons situasi dan kondisi yang ada di dunia nyata. Sedangkan dari situs resminya, Arduino didefinisikan sebagai sebuah *platform* elektronik yang *open source*, berbasis pada *software* dan *hardware* yang fleksibel dan mudah digunakan sehingga Arduino sangat membantu sebagai sebuah *board* kendali karena dapat digunakan tidak perlu membuat sistem minimum sebuah mikrokontroler lagi karena Arduino dapat dihubungkan langsung ke PC [9].

Kelebihan Arduino dari *platform hardware* mikrokontroler lain adalah sebagai berikut:

- a. IDE Arduino merupakan *multiplatform* yang dapat dijalankan di berbagai sistem operasi, seperti Windows, Macintosh dan Linux.
- b. IDE Arduino dibuat berdasarkan pada IDE *processing* yang sederhana sehingga mudah digunakan.
- c. Pemrograman Arduino menggunakan kabel yang terhubung dengan *port* USB, bukan *port* serial. Fitur ini berguna karena banyak komputer yang sekarang ini tidak memiliki *port* serial.
- d. Arduino adalah *hardware* dan *software open source* artinya dapat digunakan tanpa harus membayar.

2.5.1 Lingkungan Pemrograman Arduino

Lingkungan pemrograman ini juga disebut dengan IDE, atau *Integrated Development Environment*, dengan kelebihan diantaranya mudah dalam penggunaan IDE-nya karena kesederhanaannya, selain itu pemrograman Arduino menggunakan kabel yang terhubung dengan *port* USB, karena banyak komputer yang sekarang ini tidak memiliki *port* serial, berbeda halnya dengan sistem minimum lainnya yang membutuhkan *downloader* tersendiri untuk memasukkan program ke dalam mikro.



Gambar 2.13 Tampilan IDE Arduino

Gambar 2.13 merupakan tampilan dari *software* Arduino dan terdapat tiga bagian utama dalam bahasa pemrograman Arduino sebagai berikut.

2.5.2 Struktur Program Arduino

Bagian struktur program Arduino ini meliputi kerangka program, sintaks program, kontrol aliran program, dan operator.

a. Kerangka Program







Kerangka program Arduino sangat sederhana, yaitu terdiri atas dua blok. Blok yang pertama adalah **void setup()** yang berisi kode program yang hanya dijalankan sekali sesaat setelah Arduino dihidupkan atau di-*reset*, dan blok kedua adalah **void loop()** yang berisi kode program yang dijalankan terus menerus yang merupakan tempat untuk program utama.

b. Sintaks Program

Blok **void setup()**, **void loop()** maupun **function** harus diberi tanda kurung kurawal buka “{” sebagai tanda awal program di blok itu dan kurung kurawal tutup “}” sebagai tanda akhir program. Tanda kurung kurawal tersebut juga digunakan pada blok kontrol program, seperti **if**, **if-else**, **for-loop**, **while-loop**, dan **do-while-loop**. Untuk menandai akhir sebuah baris kode program digunakan tanda titik koma “;”. Kurangnya tanda kurung kurawal buka dan kurawal tutup ataupun titik koma

akan menyebabkan *compile error*. Untuk lebih jelasnya fungsi IDE Arduino ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Fungsi IDE Arduino

| No. | Tombol | Nama | Fungsi |
|-----|---|-----------------------|--|
| 1. |  | <i>Verify</i> | Menguji apakah ada kesalahan pada program atau <i>sketch</i> . Apabila <i>sketch</i> sudah benar, maka akan dikompilasi. Kompilasi adalah proses mengubah kode program dalam kode mesin. |
| 2. |  | <i>Upload</i> | Mengirimkan kode mesin hasil kompilasi ke <i>board</i> Arduino. |
| 3. |  | <i>New</i> | Membuat <i>sketch</i> atau lembar kerja halaman baru. |
| 4. |  | <i>Open</i> | Berfungsi untuk membuka halaman kerja yang sudah ada. |
| 5. |  | <i>Save</i> | Berfungsi untuk menyimpan halaman kerja (<i>sketch</i>). |
| 6. |  | <i>Serial Monitor</i> | Menampilkan data yang dikirim dan diterima melalui komunikasi serial. |

Sumber: DataSheet Arduino Uno

c. Kontrol Aliran Program

Kontrol aliran program ini meliputi instruksi-instruksi yang digunakan untuk membuat percabangan dan perulangan. Instruksi percabangan diantaranya adalah **if**, **if-else**, **switch-case**, **break**, **continue**, **return**, dan **goto**. Sedangkan instruksi perulangan diantaranya adalah **for-loop**, **while-loop**, **do-while-loop**. Instruksi **if** dan **if-else** akan menguji apakah kondisi tersebut dipenuhi atau tidak. Jika tidak dipenuhi maka instruksi berikutnya akan dilompati, tetapi jika dipenuhi maka instruksi tersebut akan dijalankan. Instruksi **return** digunakan untuk menghentikan proses sebuah blok **Function** dan kembali ke program utamanya, sambil membawa hasil proses apabila blok **Function** tersebut menghasilkan data. Instruksi **goto** sama seperti instruksi **break** yaitu digunakan untuk melompat keluar dari perulangan, hanya bedanya lokasi lompatan **goto** bisa diatur dengan cara menempatkan labelnya pada lokasi yang diinginkan.

2.5.3 Variabel dan Tipe Data Arduino

Berikut ini hal-hal yang berkaitan dengan variabel dan tipe data pada program Arduino:

- Sebuah variabel digunakan untuk menyimpan sebuah nilai.
- Sebuah variabel bisa diberi nama apa saja asalkan tidak sama dengan salah satu instruksi atau kode program di Arduino.
- Sebuah variabel harus dideklarasikan terlebih dulu sebelum digunakan. Mendeklarasikan variabel ini berarti menentukan tipe data variabel dan juga memberi nilai awal (opsional) pada variabel.
- Tempati dimana variabel dideklarasikan menentukan lingkup variabel. Variabel yang dideklarasikan sebelum **void setup()** disebut sebagai variabel global.
- Setiap nilai didalam array dapat dipanggil dengan nama array dan nomor indeks nilai tersebut yang mana nomor indeks dimulai dari 0 dan tipe data variabel diperlihatkan pada Tabel 2.2.

Tabel 2.2 Keterangan Tipe Data Variabel

| Tipe Data | Ukuran Data | Jangkauan Nilai |
|----------------|-------------|---|
| Boolean | 1 bit | True/False |
| Char | 8 bit | -128 s.d. 127 |
| Byte | 8 bit | 0 s.d. 255 |
| Int | 16 bit | -32768 s.d. 32767 |
| Word | 16 bit | 0 s.d. 65535 |
| Long | 32 bit | -2147483648 s.d. 2147483647 |
| Float | 32 bit | -3,4028235x10 ³⁸ s.d. 3,4028235x10 ³⁸ |

Sumber: DataSheet Arduino Uno

2.5.4 Fungsi

Bagian fungsi ini meliputi fungsi *input output* digital dan analog, advanced I/O, fungsi waktu, fungsi matematika (termasuk *random*, instruksi byte dan bit) serta fungsi komunikasi. Untuk kaki *output* analog, sebenarnya Arduino tidak memiliki kaki *output* analog. Namun dengan metode PWM (*Pulse Width Modulation*) maka beberapa kaki I/O digital Arduino dapat digunakan untuk menghasilkan sinyal analog. Berbeda dengan *input output* digital yang harus diatur

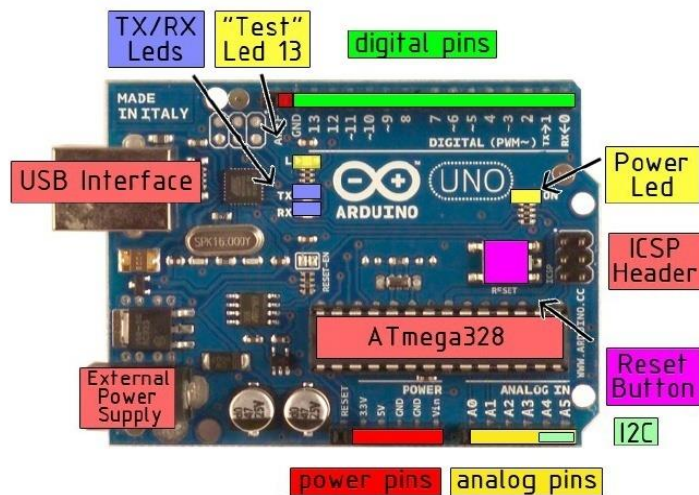
fungsi sebagai INPUT atau sebagai OUTPUT dengan instruksi **pinMode()**, maka pada *input output* analog ini tidak diperlukan pengaturan seperti itu, karena kaki input dan outputnya terpisah. Fungsi komunikasi digunakan untuk berkomunikasi dengan komputer melalui *port* serial. Kaki Arduino yang digunakan untuk fungsi ini adalah kaki D₀ (RX) dan kaki D₁ (TX).

Salah satu kelebihan Arduino adalah dapat berinteraksi dengan perangkat lunak lain seperti LabVIEW melalui komunikasi serial. Untuk membuka jendela serial monitor ada beberapa syarat sebagai berikut.

- a. Arduino harus terhubung dengan komputer.
- b. Menekan tombol bergambar kaca pembesar yang terletak dipojok paling kanan pada Toolbar Arduino.
- c. Saluran serial port yang digunakan Arduino harus benar. Pemilihan saluran ini dapat dilakukan dengan membuka menu **Tools**, dan kemudian **Serial Port**.
- d. Kecepatan komunikasi (*baudrate*) pada serial monitor harus sama dengan *baudrate* pada program Arduino. Pada umumnya kecepatan komunikasi digunakan sebesar 9600 bps.

2.5.5 Board Arduino Uno (Arduino 2010)

Arduino Uno adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega8U2 sebagai converter USB ke serialnya. “*UNO*” berarti satu dalam bahasa Italia, dan dinamai untuk menandakan keluaran (produk) Arduino. Arduino Uno mempunyai 14 pin digital *input/output* (6 diantaranya dapat digunakan sebagai *output* PWM), 6 input analog, sebuah osilator kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, ICSP *header*, dan sebuah tombol *reset*. Arduino Uno memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC dan untuk gambar fisik dari *board* Arduino Uno dapat dilihat pada Gambar 2.14.



Gambar 2.14 Board Arduino Uno

Berikut beberapa spesifikasi dari Board Arduino Uno.

Tabel 2.3 Spesifikasi Arduino Uno

| No. | Kategori | Keterangan |
|-----|---------------------------------------|----------------|
| 1. | Mikrokontroler | ATMega328 |
| 2. | Tegangan operasi | 5 Volt |
| 3. | Tegangan input (<i>recommended</i>) | 7-12 Volt |
| 4. | Batas tegangan input | 6-20 Volt |
| 5. | Digital I/O Pins | 14 (6 pin PWM) |
| 6. | Analog Input Pins | 6 |
| 7. | Arus DC tiap pin I/O | 40 mA |
| 8. | Arus DC untuk pin 3.3 | 50 mA |
| 9. | Memori Flash | 32 KB |
| 10. | SRAM | 2 KB |
| 11. | EEPROM | 1 KB |
| 12. | <i>Clock Speed</i> | 16 MHz |

Sumber: DataSheet Arduino Uno

Tabel 2.3 merupakan spesifikasi Arduino Uno, karakteristik ataupun spesifikasi dari Arduino yang digunakan perlu untuk diketahui sebelum

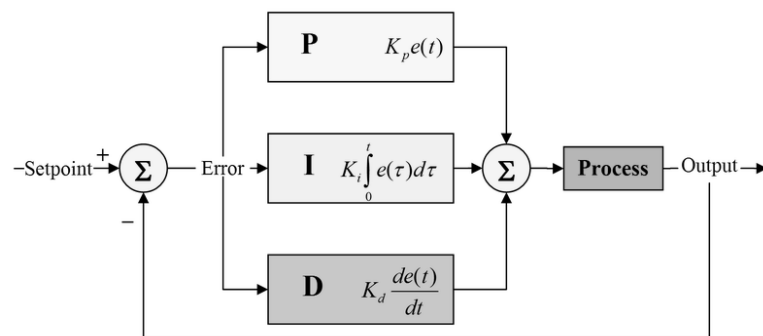
diaplikasikan, karena Arduino dapat bekerja jika dioperasikan sesuai dengan prosedur atau standar yang telah ditetapkan oleh pabrik pembuat. Arduino memiliki *datasheet* tersendiri yang berisi informasi spesifikasi Arduino.

Arduino Uno berbeda dari semua *board* Arduino sebelumnya, karena tidak menggunakan *chip driver* FTDI USB to serial. Revisi 2 dari *board* Arduino mempunyai sebuah resistor yang menarik garis 8U2 ke *ground*, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode. Revisi 3 dari *board* Arduino Uno memiliki fitur-fitur baru sebagai berikut [9]:

- a. Pinout 1.0: ditambah pin SDA dan SCL yang dekat dengan pin AREF dan dua pin baru lainnya yang diletakkan dekat dengan pin RESET, IOREF yang memungkinkan *shield* untuk menyesuaikan tegangan yang disediakan dari *board*. Untuk selanjutnya, *shield* akan dijadikan *compatible* dengan *board* yang menggunakan AVR yang beroperasi dengan tegangan 5 volt.
- b. Rangkaian RESET yang lebih andal.
- c. ATmega 16U2 menggantikan 8U2.

2.6. Pengendali PID

Pada penelitian ini pengendali proporsional – integral – derivatif digunakan karena merupakan sebuah mekanisme kontrol *loop* umpan balik yang dipergunakan secara luas dalam industri. Salah satu tugas pengendali PID adalah mereduksi sinyal kesalahan, yaitu perbedaan antara sinyal *setting* dan sinyal aktual. Semakin cepat respon pengendali dan semakin kecil nilai kesalahan yang didapat maka semakin baik pula pengendali PID tersebut. Blok diagram pengendalai PID diperlihatkan pada Gambar 2.15. dimana setiap persamaan dari P, I, D saling berhubungan untuk mendapatkan *output* yang diinginkan.



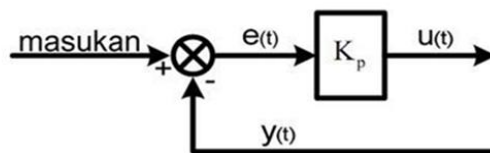
Gambar 2.15. Blok Diagram Pengendali PID

2.6.1 Kontrol Proporsional

Kontrol proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal *error*), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara *input* kontroler $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada Persamaan 2.15.

$$u(t) = K_p e(t) \quad (2.15)$$

K_p adalah konstanta proporsional. Diagram blok kontrol proporsional ditunjukkan pada Gambar 2.16.

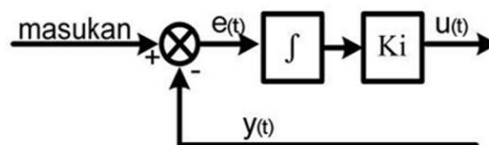


Gambar 2.16 Diagram Blok Proporsional

2.6.2 Kontrol Integral

Kontrol integral pada prinsipnya bertujuan untuk menghilangkan kesalahan keadaan tunak (*offset*) yang biasanya dihasilkan oleh kontrol proporsional. Hubungan antara *output* kontrol integral $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada Persamaan 2.16 dan diagram bloknya pada Gambar 2.17.

$$u(t) = K_i \int_0^t e(t) dt \quad (2.16)$$



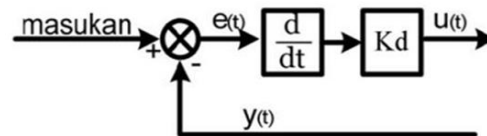
Gambar 2.17 Diagram Blok Integral

2.6.3 Kontrol Derivatif (Turunan)

Kontrol derivatif dapat disebut pengendali laju, karena *output* kontroler sebanding dengan laju perubahan sinyal *error*. Hubungan antara *output* derivatif

$u(t)$ dengan sinyal *error* $e(t)$ terlihat pada Persamaan 2.17 dan blok diagram pada Gambar 2.18.

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.17)$$



Gambar 2.18. Diagram Blok Derivatif

2.6.4 Persamaan PID

Gabungan dari proporsional, integral dan derivatif dapat dinyatakan pada persamaan 2.18 [11].

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.18)$$

Dengan:

$u(t)$ = sinyal *output* pengendali PID

K_p = Konstanta Proporsional

T_i = waktu integral

T_d = waktu derivatif

K_i = konstanta Integral (k_p/T_i)

K_d = konstanta derivatif ($K_p.T_d$)

$e(t)$ = sinyal *error* = referensi – keluaran plant = *set point* – nilai sensor

2.7. Metode Ziegler Nichols & Good Gain Method

Metode Ziegler Nichols dan *good gain method* merupakan salah satu metode selain dari perhitungan matematika yang digunakan untuk mendapatkan nilai parameter PID yang sesuai untuk sistem. Metode ini melakukan percobaan pendekatan secara pengujian untuk mendapatkan parameter PID. Terdapat 2 metode Ziegler Nichols ini yang digunakan pada sistem *loop* terbuka dan sistem *loop* tertutup yaitu metode respon langkah dan metode respon frekuensi.

2.7.1 Karakteristik Proporsional

Parameter proporsional memiliki beberapa karakteristik saat dilakukan pengujian yaitu sebagai berikut:

1. Jika nilai K_p kecil, kontroler proporsional hanya melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
2. Jika nilai K_p dinaikan, respon sistem menunjukkan semakin cepat mencapai keadaan mantapnya.
3. Jika nilai K_p diperbesar hingga mencapai nilai yang berlebih, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berosilasi.

2.7.2 Karakteristik Integral

Parameter integral mempunyai beberapa karakteristik berikut ini:

1. Keluaran kontrol membutuhkan selang waktu tertentu, sehingga kontroler integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran kontroler akan tetap pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan.

2.7.3 Karakteristik Derivatif

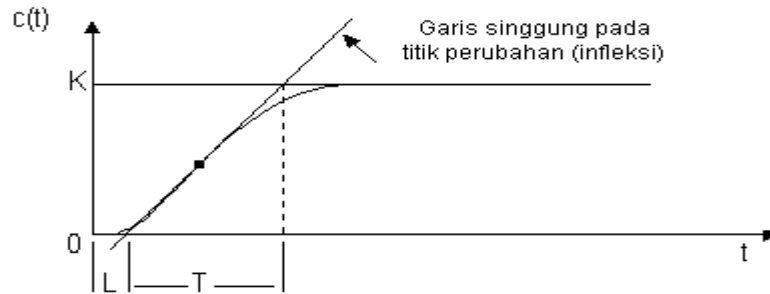
Karakteristik parameter derivatif sebagai berikut:

1. Kontroler ini tidak dapat menghasilkan keluaran bila tidak ada perubahan pada masukanya/ sinyal kesalahan.
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang akan dihasilkan kontroler tergantung pada laju perubahan sinyal kesalahan.
3. Kontroler diferensial dapat menghasilkan koreksi terhadap sinyal kesalahan.

2.7.4 Ziegler-Nichols Step Response Method

Metode ini digunakan pada percobaan *tuning open-loop*. Dimana, metode ini memiliki 2 parameter yaitu L (waktu mati) dan T (waktu tunda). Pada Gambar 2.19 terlihat garis yang bersinggungan dengan garis kurva. Garis itu akan berpotongan dengan sumbu absis dan garis maksimum. Perpotongan garis

singgung dengan sumbu absis merupakan ukuran waktu mati, dan perpotongan dengan garis maksimum merupakan waktu tunda yang diukur dari titik waktu L [12].



Gambar 2.19 Kurva Respons Langkah

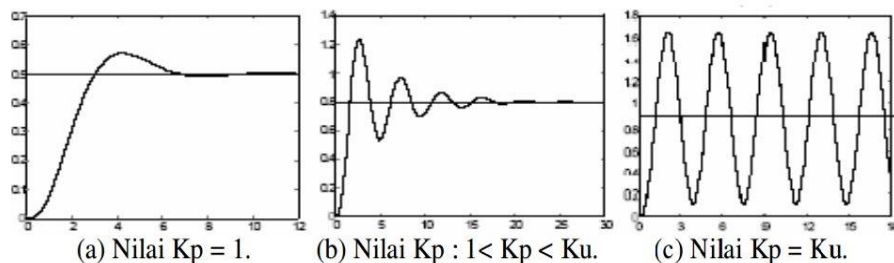
Tabel 2.4 merupakan tabel rumusan penalaan parameter PID berdasarkan kurva respon.

Tabel 2.4 Penalaan Parameter PID dengan Kurva Respon Langkah

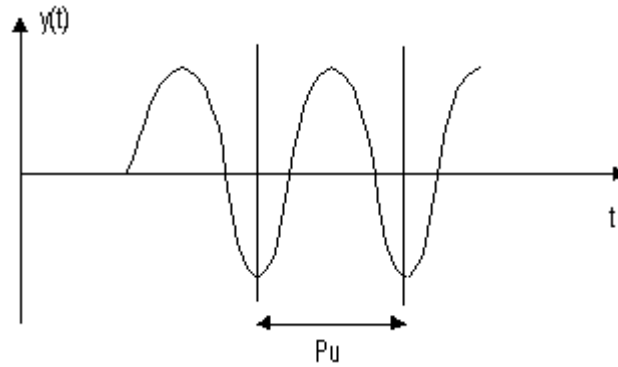
| Tipe Kontroler | K_p | T_i | T_d |
|----------------|-----------|---------|--------|
| P | T/L | \sim | 0 |
| PI | $0,9 T/L$ | $L/0,3$ | 0 |
| PID | $1,2 T/L$ | $2L$ | $0,5L$ |

2.7.5 Ziegler-Nichols Frequency Response Method

Metode respon frekuensi ini digunakan pada sistem *loop* tertutup. Metode ini diaplikasikan dengan cara besaran parameter integral diatur tak berhingga dan parameter derivatif diatur nol ($T_i = \infty$, $T_d = 0$). Parameter proporsional dinaikan secara bertahap dimulai dari nol hingga mencapai keadaan sistem yang berosilasi. Nilai penguatan proporsional pada saat sistem mencapai kondisi *sustain oscillation* disebut *ultimate gain* (K_u). Periode dari *sustained oscillation* disebut *ultimate period* (T_u).



Gambar 2.20 Karakteristik Keluaran dengan penambahan K_p



Gambar 2.21 *Ultimate Period* (T_u)

Gambar 2.20 menunjukkan perubahan gelombang atau kurva yang disebabkan oleh penambahan nilai proporsional hingga mencapai sistem keadaan berosilasi. Gambar 2.21 menunjukkan besar periode gelombang pada saat sistem mengalami osilasi [13].

Parameter PID pada metode respon frekuensi dapat dirumuskan pada tabel 2.5.

Tabel 2.5 Penalaan Parameter PID pada Respon Frekuensi

| Tipe Kontroler | K_p | T_i | T_d |
|----------------|------------|-------------------|-------------|
| P | $0,5 K_u$ | | |
| PI | $0,45 K_u$ | $\frac{1}{2} P_u$ | |
| PID | $0,6 K_u$ | $0,5 P_u$ | $0,125 P_u$ |

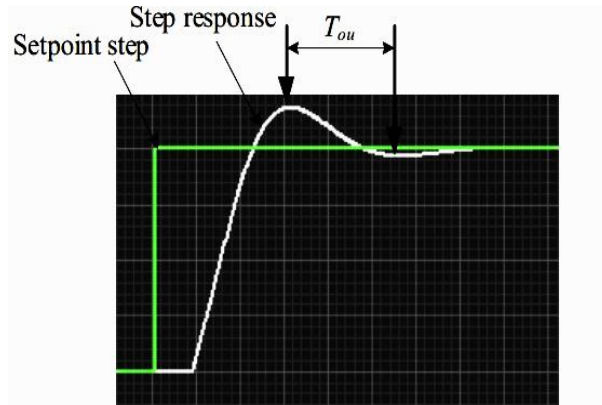
2.7.6 *Good Gain Method*

Metode *good gain* merupakan salah satu cara untuk mendapatkan parameter PID yang sesuai dengan sistem. Metode ini berbeda dengan metode Ziegler Nichols dimana pada metode ini tidak mengatur sistem kontrol sampai pada keadaan berosilasi. Adapun prosedur yang harus dilakukan dalam metode ini untuk menentukan parameter PID yaitu sebagai berikut:

1. Proses pertama dilakukan pada sistem *loop* tertutup dimana sinyal kontrol dimasukkan secara manual.
2. Kontrol proporsional dimasukan secara bertahap dimulai dari 0 atau 1 ($K_p = 0$), dan untuk waktu integral dan derivatif diset ∞ dan 0 (set $T_i = \infty$ dan $T_d = 0$). Penambahan atau pengurangan nilai K_p berpengaruh terhadap respon *overshoot* suatu sistem.
3. Set waktu integral (T_i) sesuai Persamaan 2.19.

$$T_i = 1,5 T_{ou} \quad (2.19)$$

Dimana T_{ou} adalah waktu respon suatu sistem setelah mengalami *overshoot* menuju keadaan *steady state* (*step setpoint*). Ditunjukkan pada Gambar 2.22.



Gambar 2.22 Respon T_{ou}

4. Cek kestabilan sistem kontrol setelah digunakan langkah *set point* karena ini merupakan permulaan untuk parameter integral. Parameter integral sangat mempengaruhi parameter proporsional. Jika pada kontrol PI kestabilan sistem menjadi buruk maka langkah yang harus dilakukan adalah mengurangi K_p 80% dari harga aslinya.
5. Jika ingin memasukan parameter derivatif agar kontroler menjadi PID kontrol, maka *setting* T_d sesuai dengan Persamaan 2.20 [13].

$$T_d = T_i/4 \quad (2.20)$$

BAB 3

METODE PENELITIAN

3.1. Waktu Pengujian

Penelitian dilaksanakan di Universitas Bengkulu yang bertempat di Gedung Laboratorium Fakultas Teknik. Waktu pelaksanaan dilakukan pada bulan November 2013.

3.2. Objek Penelitian

Objek penelitian yang digunakan adalah motor induksi satu fasa. Berikut adalah tabel spesifikasi untuk motor induksi satu fasa (Tabel 3.1).

Tabel 3.1. Data Spesifikasi Motor Induksi Satu Fasa

| No. | Parameter | Nilai | Satuan |
|-----|-------------------|------------------|--------|
| 1. | Tegangan nominal | 220 | Volt |
| 2. | Arus nominal | 1,9 | Ampere |
| 3. | Kecepatan (n) | 3000 | Rpm |
| 4. | Daya | $\frac{1}{4}$ Hp | Watt |
| 5. | Frekuensi | 50 | Hz |
| 6. | Kutub | 2 | |

3.3. Metode Perancangan

Metode perancangan alat sistem pengendali kecepatan motor induksi satu fasa terhadap perubahan beban dalam penelitian ini akan dilakukan dengan merancang sistem *hardware* yang meliputi 1 unit kendali mikrokontroler Arduino. Arduino disini sebagai sistem minimum yang akan mengendalikan dan memproses semua kerja alat dari *input* sampai dengan *output* yang dihasilkan. Kemudian kecepatan motor induksi satu fasa ini akan dikontrol oleh mikrokontroler menggunakan pengendali PID, dan rangkaian kontrol ini mengatur sudut penyalan frekuensi pada mosfet yang dapat mengatur putaran motor induksi sesuai dengan *input* yang diberikan berupa nilai *error* dari kecepatan referensi yang telah ditetapkan akibat terdapatnya perubahan beban. Kemudian tahapan akhir adalah

sistem *software* atau perancangan perangkat lunak untuk mikrokontroler sebagai sistem pengendali PID dengan Arduino, dan akan dilakukan pengujian secara keseluruhan.

3.3.1. Alat dan Bahan

Perancangan sistem pengendali kecepatan motor induksi satu fasa terhadap perubahan beban melalui pengaturan frekuensi dengan pengendali PID adalah suatu alat pengendali yang terprogram menggunakan pengendali PID untuk mengatur kecepatan motor terhadap perubahan beban sehingga mampu menjaga kecepatan motor induksi pada kecepatan referensi yang telah ditentukan. Alat ini juga mampu memberikan nilai putaran yang diubah melalui input frekuensi yang diberikan dan akan menghasilkan putaran sesuai dengan besar frekuensinya. Alat yang dibutuhkan dalam perancangan dan pembuatan alat ini adalah :

1. Penyearah gelombang penuh menggunakan trafo *center tap*
Penyearah gelombang penuh menggunakan 2 buah dioda berkapasitas 5 ampere dan trafo *center tap* (CT) berkapasitas 5 ampere.
2. Filter kapasitor
Filter menggunakan kapasitor elektrolit atau *elco* bertipe 4700 μf 50 v.
3. Inverter (PWM)
Inverter yang digunakan adalah MOSFET IRF9630 dan MOSFET IRF630 yang memiliki spesifikasi VDS maksimal -200 volt dan 200 volt, , *gate source voltage* +/- 20 v.
4. Motor induksi satu fasa
Motor induksi satu fasa yang digunakan adalah motor induksi yang memiliki kapasitas tegangan sumber 220 volt, 3000 RPM, 1,9 A.
5. Fotodioda dan fototransistor
Fotodioda dan fototransistor yang digunakan berukuran 3 mm.
6. Sistem minimum mikrokontroler Arduino uno
7. LCD karakter 16x2
8. *Trafo step down* berkapasitas 10 ampere
9. Rangkaian
10. Kabel konektor

3.3.2. Tahapan Penelitian

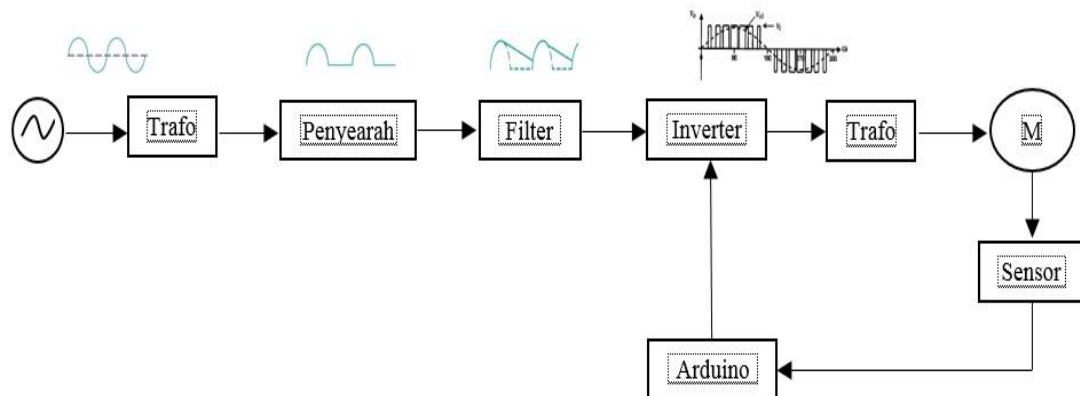
Tahap pertama yaitu perancangan *hardware* berupa mikrokontroler yang berfungsi sebagai pusat pengendali beberapa *hardware* yang telah terhubung, diantaranya adalah fotodioda yang digunakan untuk membaca kecepatan motor induksi, LCD, *keypad* dan mosfet yang berguna sebagai pengubah tegangan dari DC ke AC dan juga sebagai tempat *switch* frekuensi menggunakan PWM. Tahap kedua adalah perancangan *software*. Menerapkan pengendali PID pada Arduino yang digunakan pada mikrokontroler untuk menstabilkan putaran motor induksi satu fasa melalui pengaturan frekuensi terhadap perubahan beban. Tahap selanjutnya yaitu melakukan analisa dan pembahasan yang akan disesuaikan dengan hasil pengujian yang diperoleh nantinya. Tahap terakhir adalah menyimpulkan hasil tahapan-tahapan yang telah dijelaskan sebelumnya. Hasil akhir dari penelitian ini adalah perancangan pengaturan kecepatan motor induksi satu fasa dengan PWM menggunakan pengendali PID berbasis Aduino.

3.4. Perancangan *Hardware*

Hardware merupakan alat atau komponen utama sebagai pengendali kecepatan dan putaran motor induksi satu fasa yang terdiri dari beberapa bagian blok masing-masing yang memiliki fungsi kerja yang penting dan saling mendukung satu sama lainnya.

Blok diagram pada Gambar 3.1 merupakan alat sebuah pengendali kecepatan motor induksi satu fasa. Perencanaan perangkat keras ini meliputi sistem konverter daya yang terdiri dari penyearah gelombang penuh, filter kapasitor, dan inverter berjenis mosfet untuk mengubah tegangan DC ke tegangan AC sekaligus sebagai tempat pengaturan frekuensi dengan modulasi lebar pulsa. Pada pengendali menggunakan mikrokontroler Arduino menggunakan pengendali PID untuk mengatur frekuensi pada inverter yang digunakan.

Diagram blok ini menjelaskan cara kerja dari pembuatan pengendali kecepatan motor induksi satu fasa ini. Berikut adalah diagram blok perancangan dan pembuatan sistem pengendali kecepatan putaran motor induksi satu fasa.



Gambar 3.1. Diagram blok sistem yang diusulkan

Dari blok diagram tersebut, dapat dijelaskan prinsip kerja dari diagram tersebut yaitu sumber daya PLN yang merupakan tegangan bolak balik satu fasa disearahkan menggunakan penyearah dioda jembatan penuh *center tap* untuk menghasilkan tegangan searah. Sebuah filter kapasitor ditempatkan pada keluaran penyearah untuk menghasilkan tegangan searah yang lebih halus. Untuk menghasilkan tegangan bolak-balik sebagai input motor induksi digunakan inverter satu fasa jembatan penuh menggunakan sakelar MOSFET (*Metal Oxide Silicon Field Effect Transisto*). Rangkaian penyearah dan inverter menggunakan jembatan penuh dipilih karena mampu menghasilkan transfer daya yang efisien dan penggunaan tegangan yang lebih optimal dibanding rangkaian setengah jembatan. Metode yang digunakan untuk memodulasi inverter adalah *pulse-width modulation* (PWM).

Untuk mendapatkan kecepatan putaran yang diharapkan digunakan sistem kontrol dengan pengendali PID. Pengendali PID merupakan pengendali yang mudah untuk diterapkan dan berbiaya rendah. Photodiode digunakan sebagai sensor untuk membaca kecepatan putaran motor. Kecepatan referensi dibandingkan dengan kecepatan yang terukur diumpkan kepada pengendali sebagai sinyal *error*. Pengendali mengambil keputusan berdasarkan *error* yang terbaca dengan mengirimkan sinyal kepada inverter SPWM untuk membangkitkan frekuensi yang tepat.

3.4.1. Arduino

Rangkaian sistem minimum yang digunakan adalah rangkaian Arduino uno yang didalamnya terdapat mikrokontroler ATmega328. Sistem minimum ini adalah sistem yang mengatur keseluruhan perintah untuk mendapatkan hasil yang diinginkan. Dimana, pengaturannya berupa perintah untuk mengoperasikan MOSFET dalam mengatur lebar pulsa untuk mendapatkan frekuensi yang dapat mengatur kecepatan motor induksi tersebut. Metode pengendali yang dikonversikan pada Arduino adalah pengendali PID. Contoh pengkodean PID pada Arduino diperlihatkan pada Gambar 3.2.

```
PID:
    Error = Setpoint - Actual
    Integral = Integral + (Error*dt)
    Derivative = (Error - Previous_error)/dt
    Drive = (Error*kP) + (Integral*kI) + (Derivative*kD)
    Previous_error = Error
    wait(dt)
GOTO PID

//----- Calculates the PID drive value -----
Actual = analogRead(Position);
Error = SetPt - Actual;

if (abs(Error) < IntThresh){           // prevent integral 'windup'
    Integral = Integral + Error;       // accumulate the error integral
}
else {
    Integral=0;                       // zero it if out of bounds
}
P = Error*kP;                         // calc proportional term
I = Integral*kI;                     // integral term
D = (Last-Actual)*kD;                // derivative term
Drive = P + I + D;                   // Total drive = P+I+D
Drive = Drive*ScaleFactor;           // scale Drive to be in the range 0-255
if (Drive < 0){                       // Check which direction to go.
    digitalWrite (Direction,LOW);    // change direction as needed
}
else {
    digitalWrite (Direction,HIGH);   // depending on the sign of Error
}
if (abs(Drive)>255) {
    Drive=255;
}
analogWrite (Motor,Drive);           // send PWM command to motor board
Last = Actual;                       // save current value for next time
}
```

Gambar 3.2 Pengkodean PID pada Arduino

3.4.2. Inverter Daya

Objek penelitian untuk inverter daya menggunakan saklar MOSFET (*Metal Oxide Silicon Field Effect Transistor*) tipe IRF9630 dan IRF630. Inverter ini digunakan selain untuk mengubah tegangan DC menjadi tegangan AC sebagai input motor induksi, inverter ini juga digunakan untuk mengatur lebar pulsa demi

mendapatkan frekuensi yang digunakan untuk mengatur kecepatan motor induksi. Cara yang digunakan untuk mengatur lebar pulsa dengan mengatur sinyal *on/off* pada gate MOSFET tersebut.

3.4.3. Optocoupler

Sensor yang digunakan pada penelitian ini adalah sensor *optocoupler*. *Optocoupler* merupakan sensor yang terdiri dari fotodioda dan fototransistor yang berukuran 3mm. *Optocoupler* ini diletakan didekat poros rotor untuk melihat atau mendeteksi putaran motor dengan cara membaca perbedaan warna gelap dan terang pada poros rotor. Gelap dan terang didapat dengan memodifikasi besi poros rotor dengan membuat garis – garis hitam dan putih. Hasil pembacaan dari sensor ini akan dikirim sebagai input masukan pada Arduino.

3.4.4. LCD

Pada penelitian ini digunakan LCD berukuran 16x2, dimana LCD tersebut berfungsi untuk menampilkan besaran kecepatan motor ketika ditambah atau dikurangi beban agar dapat mengetahui apakah kecepatan motor berubah dari kecepatan referensinya. Besaran ini didapat dari proses pengolahan data pada Arduino.

3.5. Perancangan Software

3.5.1 Perancangan *Proportional, Integral, Derivatif* (PID)

Pada perancangan sistem kendali ini sangat dipengaruhi oleh parameter proporsional, integral, dan derivatif untuk mendapatkan sistem kontrol yang sesuai. Semakin baik nilai parameter PID maka sistem kendali akan semakin cepat mengubah sistem pada keadaan *set point* yang telah ditetapkan walaupun terdapat sinyal *error* yang berubah-ubah. Untuk mendapatkan parameter proporsional, integral dan derivatif digunakan metode *try & error*, metode Ziegler Nichols dan metode *good gain*. Parameter PID didapat dimulai dari mengikuti langkah – langkah prosedur pada metode *good gain* dan dilakukannya perhitungan. Setelah didapat data perhitungan, selanjutnya dilakukan metode *try & error* hingga didapatkan parameter PID yang sesuai untuk sistem.

3.6. Analisis Kerja

Analisis kerja dari penelitian perancangan sistem pengendali kecepatan motor induksi satu fasa menggunakan PID ini dilakukan pada saat setelah pengujian alat selesai. Analisis rangkaian kontrol motor salah satunya diperlukan untuk mengendalikan kecepatan motor. Dengan memberikan kecepatan referensi pada masukan yang akan dikonversi oleh mikrokontroler menjadi besaran frekuensi, maka akan didapatkan nilai *error* dan *delta error* antara kecepatan referensi terhadap kecepatan yang sebenarnya di motor tersebut akibat terjadinya perubahan beban yang diberikan dikarenakan pada saat diberi beban kecepatan motor akan menurun yang akan menyebabkan turunnya frekuensi hal ini karena kecepatan motor berbanding lurus dengan frekuensinya.

Analisis yang dilakukan meliputi pengujian kecepatan motor menggunakan pengendali PID sebagai penstabil kecepatan motor induksi satu fasa dalam kondisi berbeban. Pengendali PID berfungsi untuk mengontrol motor agar dapat cepat dan tetap pada posisi *steady state*. Setelah dilakukan pengujian menggunakan pengendali PID, maka akan dianalisis perbandingan antara kecepatan referensi terhadap kecepatan yang dihasilkan motor, respon waktu pada saat n_0 konstan terhadap n ref.

Untuk mendapatkan nilai konvert di saat memasukkan nilai RPM referensi ke pengendali, kecepatan dibaca oleh sensor fotodioda. Kecepatan yang dibaca oleh fotodioda akan dikirim ke mikrokontroler. Pada mikrokontroler kenaikan dan penurunan kecepatan dibaca menjadi besar kecilnya frekuensi. Hal ini akan memerintahkan MOSFET dalam mengatur sinyal *gate* untuk sudut penyalan agar didapat frekuensi dan kecepatan referensi.

3.7. Pengujian

Pengujian yang akan dilakukan terhadap *prototy* motor induksi satu fasa terdiri atas:

1. Pengujian karakteristik respon pengendali terhadap perubahan beban.

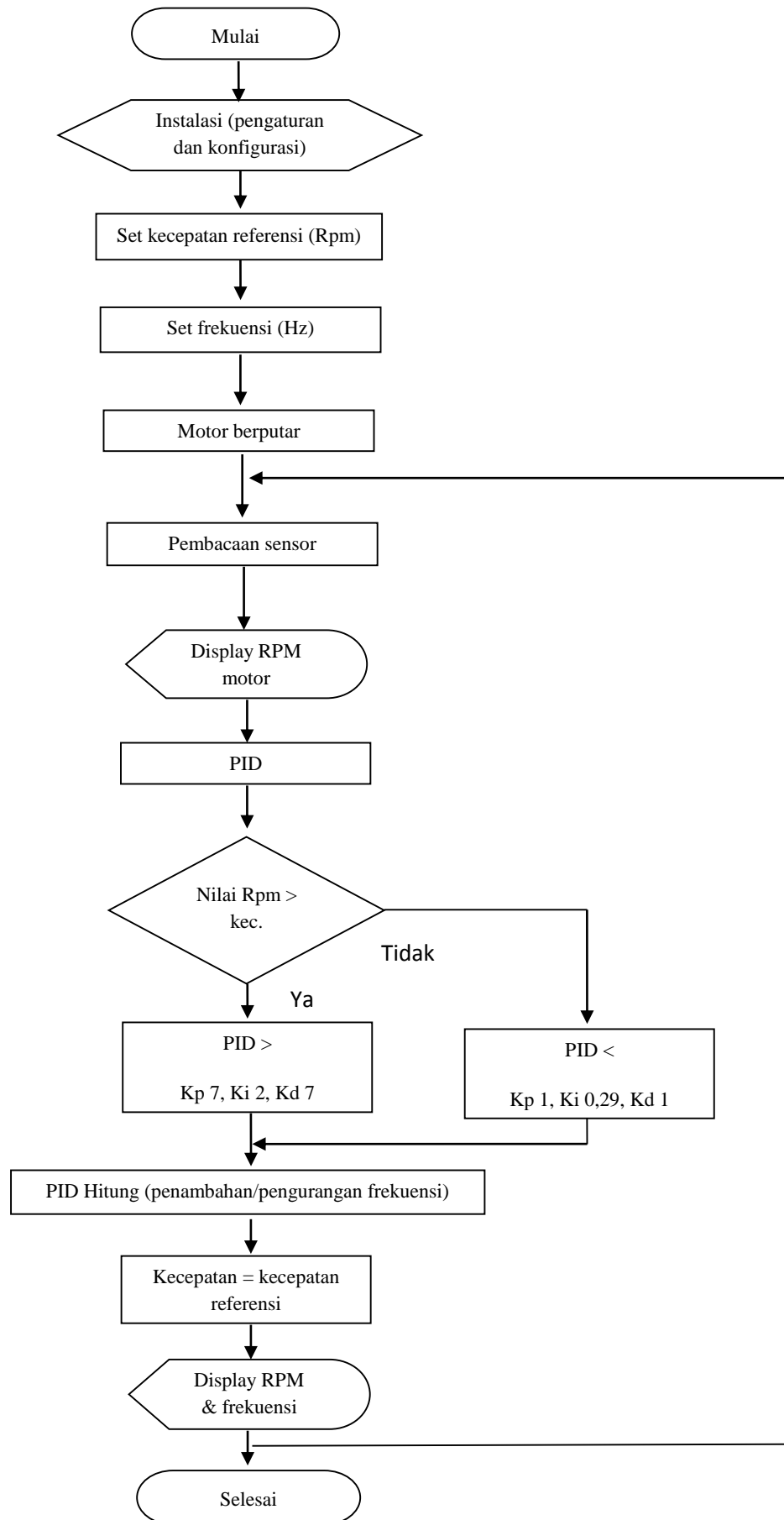
Pengujian respon pengendali terdiri atas:

- *Rise time*
- *Steady-state time*

2. Pengujian efisiensi daya
3. Pengujian konsumsi energi

3.8. Diagram Alir Sistem Kendali

Diagram alir sistem kendali pada Gambar 3.3 dapat dijelaskan bahwa proses pertama yaitu mulai persiapan berupa persiapan alat dan instalasi dalam pengaturan waktu dan konfigurasi lainnya. Proses selanjutnya atur rpm referensi yang didapat dari set frekuensi yang telah ditetapkan. sehingga dapat dikalkulasikan menjadi frekuensi referensi untuk nilai set rpm. Selanjutnya kalkulasi nilai toleransi RPM minimum dan maksimum untuk set RPM. Kemudian pembacaan motor telah stabil atau tidak, apabila telah stabil proses selanjutnya kalkulasi nilai RPM motor dari pulsa sensor yang membaca kecepatan aktual motor. Pada *display* akan menampilkan seberapa besar kecepatan motor, apabila kecepatan motor kurang dari nilai minimum toleransi maka program pengendali PID akan menaikkan nilai frekuensi dan sebaliknya apabila kecepatan motor lebih besar dari kecepatan maksimum toleransi maka pengendali akan menurunkan frekuensi motor. Langkah akhir adalah kalkulasi nilai waktu untuk menghasilkan lebar pulsa yang sesuai dengan set frekuensi yang telah ditentukan.



Gambar 3.3 Diagram alir sistem pengendali